DistilBERT Augmented with Mixture of Local and Global Experts

Stanford CS224N Default Final Project, Robust QA

Joshua Tanner, Philipp Reineke
Department of Medicine, Department of Management Science and Engineering
Stanford University
{jvtanner, preineke}@stanford.edu

Abstract

Few-shot systems are valuable because they enable precise predictions using small amounts of expensive training data, making them particularly cost-efficient. In this paper, we explore a technique to improve the few-shot question answering capabilities of a pre-trained language model. We adjust a pre-trained DistilBERT model such that it leverages datasets with large amounts of training data to achieve higher question-answering performance on datasets with very small amounts of available training data using a novel inner- and outer-layer of Mixture of Experts approach.

1 Introduction

The entrance of BERT (Bidirectional Encoder Representations form Transformers) (Vaswani, 2017) [1] onto the NLP stage caused a stir among the Machine Learning community after posting record numbers for a variety of language tasks, including Question Answering. The secret to BERT's success was largely attributable to its bidirectional training of a transformer made possible by a masked language model. This was in contrast to the typical approaches of the day which included sequence movement in either left-to-right or a combination of both left-to-right and right-to-left training methods.

However, BERT and more recent models continue to rely on a massive number of parameters, numbering in the hundreds of millions or more. This makes them costly to train and unrealistic foundations for transfer learning applications for on-device computations. In this paper, we utilize DistilBERT (Sanh, 2019) [2], a lighter and faster version of BERT which has shown to retain nearly all of BERT's language understanding abilities. Our goal is to successfully implement a novel Mixture of Experts (MoE) approach first introduced by Jacobs and colleagues (1991) [3] to dramatically raise DistilBERT's transfer learning capabilities within question answering in the context of dissimilar and scarce portions of training data.

The MoE approach saw first light thirty years ago and has since been broadly adopted in dozens of applications ranging from predicting rank data (Gormley & Murphy, 2010) [4] and time series data (Fruhwirth-Schnatter et al., 2012) [5] to longitudinal data (Tang & Qu, 2015) [6] and non-normal data (Villani et al, 2009) [7]. Many different flavors of MoE have been tried, tested, and found useful. But the core principle behind each design is to subject the model parameters to a function incorporating concomitant covariates.

2 Related Work

Perhaps the most significant application of MoE in recent years has been the contribution made by members of Google Brain (Shazeer et al., 2017) [8] who implemented a sparsely-gated 137 billion-parameter MoE layer between LSTM layers to achieve state-of-the-art performance at a lower cost. This landmark paper was the first to implement conditional computation in deep networks successfully.

Prior work focused on fleshing out various aspects of the MoE framework, such as hierarchical structure (Yao et al., 2009) [9], infinite number of experts (Rasmussen & Ghahramani, 2002) [10], and adding experts in a sequential manner (Aljundi et al., 2016) [11]. Eigen and colleagues (2013) [12] introduced the idea of using many MoE's with separate gating networks, which proved a far more powerful approach than other variations since this approach allowed for the engagement of the neural network with multiple sub-problems which are often present in machine learning applications.

While Eigen and colleagues stacked two groups of MoE's, our approach will instead be to incorporate several "inner" MoE's within "outer" MoE structures. This approach leverages both the insight of independently trained experts and specialization of experts which are guided to particularly salient aspects of our training data.

3 Approach

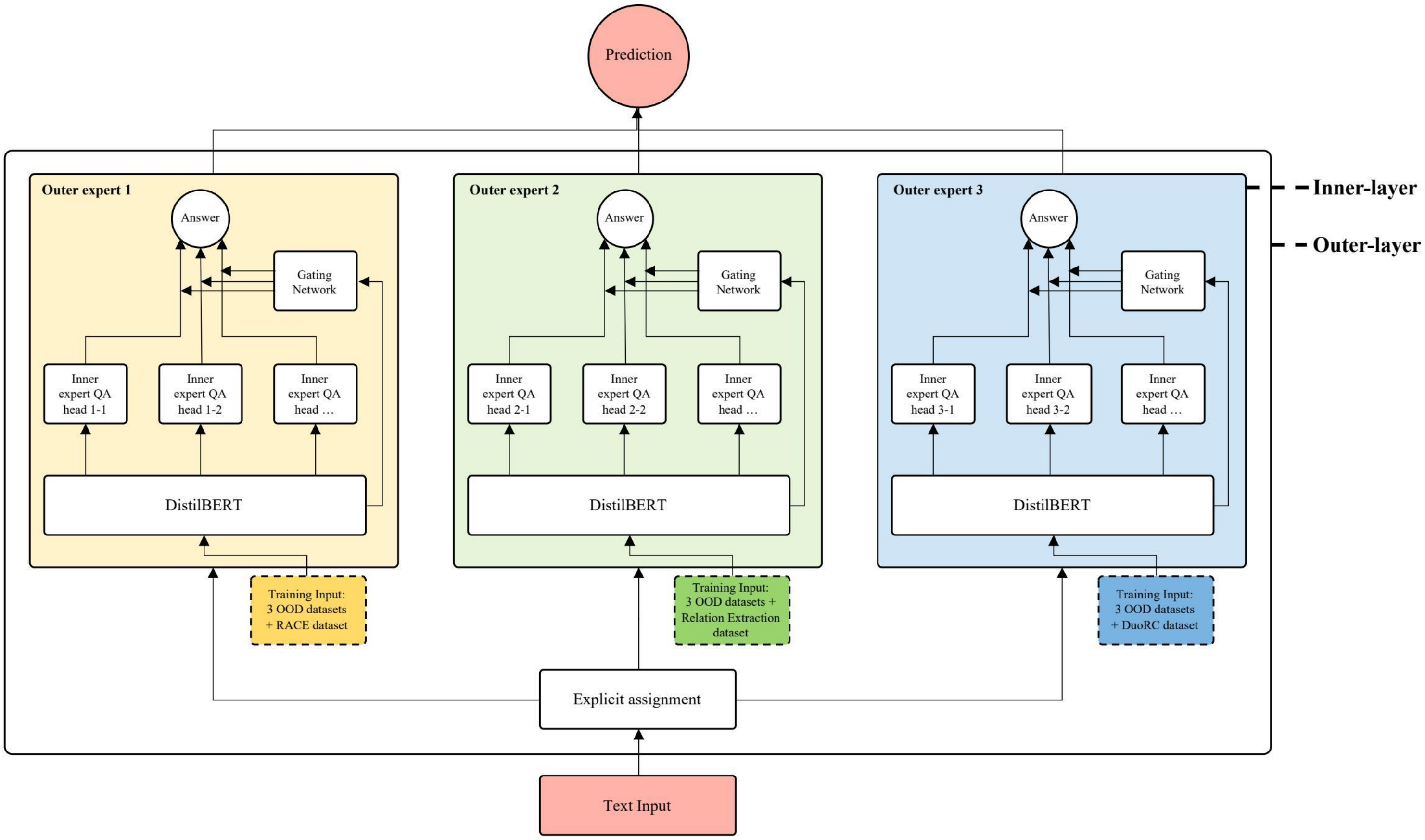


Figure 1: Model structure illustration.

Our approach consists of the implementation of MoE on two separate levels:

Outer-layer: Central to our approach is the realization that our prediction tasks consists of predictions from multiple *separate* out-of-domain (OOD) datasets. However, for each of these OOD datasets we have separate training and validation sets and, crucially, we know which data originated from which dataset. This allows us to train a separate, specialized model for data prediction from each OOD dataset. Thus, for the outer-layer of our model we employ the philosophy of the MoE method without relying on a gating network because we can manually assign data to experts based on its database of origin. By eliminating the gating component on the outermost level, we can obtain 100% assignment accuracy with 0 noise and reduce the number of total parameters that need to be trained. In summary, our outer-layer implementation is grounded in a repeatable process for combining expert models in contexts where the data origin of OOD data is known.

This approach has several benefits. First, it allows us to account for the scarcity of OOD data and train models without the confounding influence of other OOD datasets, resulting in minimal noise. Second, it allows us to train our inner-layer models in parallel on separate machines, resulting in dramatically increased training speed, a benefit of parallelization. Third, splitting our inner-layers across different machines means that each machine requires less memory since we do not have to store multiple large models in RAM. Finally, we gain additional flexibility since we can evaluate model performance of single experts separately, using different model architectures that yield optimal performance for their respective OOD dataset.

Inner-layer: For our inner-layer experts we leverage the MoE method to optimize transfer learning from our in-domain (IND) data to predict OOD observations. First, we instantiated a pre-trained 'distilbert-base-uncased' DistilBERT as a question answering model. This model contains a randomly initialized question-answering head layer which requires further finetuning to translate DistilBERT output tensors to predictions in a question-answering format. We removed this question-answering head layer and replaced it with a MoE layer consisting of 4 to 5 experts (depending on the OOD dataset) and a gating network. Each of our experts is a Multilayer Perceptron (MLP) with a hidden layer size of 64 and an output size of 2. However, we removed the final softmax activation function common in MLPs to make the experts useful as disparate question-answering heads. Rather than outputting probability distributions, these heads output two numeric values indicating the start- and stop-characters which "highlight" where the answer can be found found within the input text. Our gating network is an MLP with hidden size 64 and output size "number of expert" which drops the sequence length tensor dimension but contains a softmax activation layer. In the layer's forward function, we weight each expert's output by the probability distribution obtained from the gate, effectively deciding which of the experts we will listen to in making predictions from data considering the nature of the particular input observation.

We implement a new loss function to 1) decouple each network's weight updates from one another and 2) place each network's performance in the context of the others:

$$E^c = -log \sum_i p_i^c e^{\frac{1}{2}||\mathbf{d}^c - \mathbf{o}_i^c||^2} \tag{1}$$

Ostensibly, this means that each of our experts focuses on a different component / different components of the training data. This allows our model to predict observations from the small OOD dataset that have components similar to those that appear in the large IND data using experts with greater exposure to our large IND data. These observations will therefore lead to predictions with a higher degree of precision. Conversely, observations that have components which are different from those of our IND observations but similar to those of OOD observations can be predicted using experts that are more strongly trained on the scarce OOD training data. These predictions may be less precise due to lower data availability, but are at least not biased by dissimilar IND data training observations.

Our sequential approach of implementing a DistilBERT model with an internal MoE-layer has several distinct benefits over alternatives. First, our method adds a minimal number of additional layers to the model. Therefore, it is faster and more memory-efficient than a model with several

DistilBERT experts linked within a MoE architecture would be. Second, since the MoE gate distributes back-propagation of gradients across experts, each expert is effectively only trained on a share of the available training data. Therefore, we need to keep each expert moderately complex so as to not over-fit on the OOD data. Our design's MLP experts achieve this whereas a full DistilBERT model expert would likely overfit the OOD data. Third, while backprop is divided amongst the question-answering heads, there is no other aspect of the DistilBERT model that divides its learning. All backpropagated gradients flow through the entire DistilBERT model. This allows our DistilBERT model to be finetuned on all data, not just the data the gate allocates to a specific expert. By engineering our model this way, we allow all experts to benefit from transfer learning from other experts via the upstream DistilBERT model.

In short, our approach uses MoE both as a design principle (in the outer-layer) and as a modelling method (in the inner-layer), yielding several performance and prediction benefits. Our model strongly outperforms our baseline on the validation set as a result.

4 Experiments

4.1 Data

In-domain datasets:

- **SQuAD (SQ)** - paragraphs from Wikipedia. Around 150k Questions and answers are crowd-sourced using Amazon Mechanical Turk. About half of the 150k questions cannot be answered using the provided paragraphs. The answers to the other questions are in the form of snippets of text taken directly from the source paragraph (Rajpurkar et al., 2016) [13].

- **Natural Questions (NA)** - data set derived from questions posed to Google in which Wikipedia pages from the top 5 results constitute the input. The output consists of a long-form, short-form, or 'null' if no answer can be derived explicitly from the Wikipedia text (Kwiatkowski et al., 2019) [14].

- **NewsQA (NE)** - a machine comprehension data set of 100k human-generated Q&A pairs from 10k CNN news articles. Answers are spans of text from the corresponding input context (Trischler et al., 2016) [15].

Out-of-domain datasets:

- **RACE (RA)** - 100k reading comprehension questions and answers from 28k passages asked to Chinese teenagers. Significant portion of reasoning is involved in answering these questions, resulting in massive model vs. human performance difference in recent models (43% vs. 95%, respectively) (Lai et al., 2017) [16].

- **Relation Extraction (RE)** - dataset of 30 million question answer pairs from Wikidata and WikiReading which were curated as part of a relationship extraction project (Levy et al., 2017) [17].

- **DuoRC (DU)** - 186k Q&A pairs based on 7680 movie plots. Each question is drawn from either the IMBd or Wikipedia description of the movie with the corresponding answer drawn from the whichever description the question did not come from. This means that the answer is almost certainly not a simple extraction from the input text (Saha et al., 2018) [18].

For training our models, we do not use the complete datasets listed above but subsets thereof that are further split into train, validation, and test sets. See Appendix 1 for a table listing the number of "answers" in each of the single data-subsets.

4.2 Evaluation method

We use Exact Match (binary score that is 1 if system output and ground truth match exactly, and 0 otherwise) and F1 scores (the harmonic mean of precision and recall of single words in the system output vs. ground truth) to evaluate our results.

4.3 Experimental details, results, and results discussion

First we trained our baseline, a single DistilBERT model, to predict all OOD validation sets at once. In separate models we conducted additional finetuning to improve model performance. We used model 4, the model with the greatest overall predictive performance, as a baseline. Then we tested whether replacing QA heads with an MoE layer would yield performance improvements, which it did (see model 5), even without further finetuning.

Model number	**Training step 1**				**Training step 2**		**Eval data**	**EM**	**F1**
	Train data	**Val data**	**MoE layer (heads)**	**eval-every param**	**Train data**	**Val data**			
1	SQ NA NE	SQ NA NE	No	5000			RA RE DU	33.25	48.43
2	SQ NA NE	SQ NA NE	No	5000	RA RE DU	RA RE DU	RA RE DU	32.98	48.20
3	SQ NA NE	RA RE DU	No	5000			RA RE DU	32.46	48.53
4 (B)	SQ NA NE	RA RE DU	No	5000	RA RE DU	RA RE DU	RA RE DU	33.77	49.57
5	SQ NA NE RA RE DU	RA RE DU	Yes (6)	5000			RA RE DU	34.55	50.72

Table 1: Baseline models and MoE full model. SQ, NA, NE, RA, RE, DU abbreviations indicate datasets used. Green indicates in-domain datasets. Orange indicates out-of-domain datasets. All EM and F1 values for validation set.

Then, we optimized our inner-layer framework. To contextualize the performance of the single experts of the outer-layer, we create a second baseline consisting of the predictive performance of the baseline model applied to each OOD dataset separately.

Model number	**Training step 1**				**Training step 2**		**Eval data**	**EM**	**F1**
	Train data	**Val data**	**MoE layer (heads)**	**eval-every param**	**Train data**	**Val data**			
4 (B)	SQ NA NE	RA RE DU	No	5000	RA RE DU	RA RE DU	RA	24.22	36.02
4 (B)	SQ NA NE	RA RE DU	No	5000	RA RE DU	RA RE DU	RE	42.97	68.93
4 (B)	SQ NA NE	RA RE DU	No	5000	RA RE DU	RA RE DU	DU	34.13	43.68

Table 2: Single OOD dataset prediction baselines for Model 4. All EM and F1 values for validation set.

Using four inner-layer expert QA heads (one per training dataset) for each inner-layer model improves on this new baseline for two out of three of our datasets. To maintain a standard of comparability with our baseline models, we preserved most configurations such as the learning rate (0.00003). Loss graphs show that the training loss decrease continuously. However, validation scores displayed periodic fluctuations. To account for this, we reduced the evaluation interval which dictates checkpoint save intervals.

Lastly, we experimented with the number of experts of the inner-layer models and found that five experts (one expert per training dataset + 1) for the inner-layer models yields maximum performance in terms of EM for the RE and DU datasets.

We ran further experiments to test whether additional performance improvements could be achieved. Particularly, we tested whether freezing the DistilBERT model in our inner-layer experts by disconnecting it from the torch graph and just training the QA head MoE layer alone would yield similar training results at a faster rate. While these models did indeed train much faster than the models that back-propagated losses throughout the entire DistilBERT model (5h10m-6h53m vs. the 12h49-15h18m in models 6 to 11), their performance did not outperform the baseline. We also attempted to further finetune MoE inner-layer models, which generated no changes in predictive performance.

Model number	Training				Eval data	Train time	EM	F1	Overall EM	Overall F1
	Train data	Val data	MoE layer (heads)	eval-every param						
6	SQ NA NE RA	RA	Yes (4)	500	RA	12h49m52s	**27.34**	**43.65**	37.17	54.65
7	SQ NA NE RE	RE	Yes (4)	500	RE	12h54m19s	49.22	74.48		
8	SQ NA NE DU	DU	Yes (4)	500	DU	13h50m00s	34.92	**45.69**		
9	SQ NA NE RA	RA	Yes (5)	500	RA	14h07m16s	21.09	36.27	37.69	52.06
10	SQ NA NE RE	RE	Yes (5)	500	RE	13h52m23s	**56.25**	**75.04**		
11	SQ NA NE DU	DU	Yes (5)	500	DU	15h18m05s	**35.71**	44.77		

Table 3: MoE model single OOD dataset prediction performance and training times for 4 and 5 expert MoE layers. All EM and F1 values for validation sets. Highest scores for the particular eval dataset are highlighted in bold.

In composing our final model, we saved the validation checkpoints by highest F1 metric and then chose the best performing 4 vs 5 expert models by EM metric. Thus, we balance the impact of the two evaluation metrics on the composition of our final model. Our final model achieves a performance of EM 39.79 and F1 54.53 on the validation set and EM 41.88 and F1 59.60 on the test set.

Model number	Training				Eval data	EM (val)	F1 (val)	Overall EM (val)	Overall F1 (val)	Overall EM (test)	Overall F1 (test)
	Train data	Val data	MoE layer (heads)	eval-every param							
6	SQ NA NE RA	RA	Yes (4)	500	RA	27.34	43.65	39.79	54.54	41.88	59.60
10	SQ NA NE RE	RE	Yes (4)	500	RE	56.25	75.04				
11	SQ NA NE DU	DU	Yes (4)	500	DU	35.71	44.77				

Table 4: Inner-layer experts in final model.

This performance on the test set is partly due to the differences in observation shares from the test vs validation set databases - with shares of 9.61% RA, 61.77% RE, and 28.62% DU in the test set and 33.51% RA, 33.51% RE, and 32.98% DU in the validation set. Had we observed the same performance in the validation set for each inner-layer model given the mixture of observations in the test set, we would have expected a performance of EM 47.59 and F1 63.36 on the test set. The fact that our model's performance on the test set is well below this benchmark suggests that our inner-layer models overfit the validation data. In hindsight, we may have been able to increase our performance on the test set by increasing the "eval-every" parameter, by manually defining the step where we save the model checkpoint used for prediction (based on smoothed performance graphs reported in Appendix 2), or by defining a function that chooses the final model checkpoint based on smoothed performance values. However, we did not do this so as to not artificially optimize our results based on the test set results.

5 Qualitative analysis

Our results provide several interesting findings. The superior performance of our model compared to the baseline lends credence to our inner-outer-layer MoE model approach. However, we make the following observations that may be used for further improvements to the model in the future:

First, our results indicate a confounding vs. transfer-learning robustness trade-off between predicting multiple OOD datasets. Training on several IND and OOD datasets at once creates an opportunity for observations from one OOD dataset to inform the predictions on other OOD datasets (learning being transferred across contexts). However, training parameters for prediction in one OOD dataset may also interfere with training parameters for predictions in another OOD dataset (confounding predictions in other contexts). Our results in Model 5 show that MoE is one way to partially resolve this tradeoff - by using a sequential layer of experts where separate parameters can be trained for different datasets while still allowing learning transfer to occur. However, Models 6 through 11 show that in our context, the noise reduction benefits from performing just a single OOD prediction task per outer-layer expert outweigh the costs of diminished transfer-learning from other OOD datasets. Accordingly, Model 5 outperforms the baseline for predictions on the validation set, but does not outperform our MoE inner-outer model.

Second, we note that some of our inner-layer models perform better on four input datasets with five experts. This is surprising since we would have expected each expert to (roughly) focus on one training dataset each. The performance increase of models with five experts (number of training datasets + 1) may indicate that models benefit from adding one expert who may represent a "general text baseline" while the remaining experts may represent components unique to the different datasets that go beyond this general baseline. In this case, the gate would then weigh expert outputs as a mixture of the "general text baseline" expert plus inputs from remaining experts whose predictions correspond to components of the observation that go beyond the "general text baseline".

Third, we note that our validation performance fluctuates in our inner-layer models despite steadily declining training loss (see Appendix 2). We take these observations as an indication that (though our "eval-every" checkpoint storage parameter size is too small) our model is not generally over-fitting the training data. Instead, we believe that part of these fluctuations may be caused by our sequential use of DistilBERT and MoE layers. Changes in the DistilBERT model parameters resulting from backpropagated losses from one expert may result in changes to the output of the DistilBERT layer. These changes may in turn change the ideal distribution of components which should be handled by a given expert. Thus, certain changes in the underlying DistilBERT model may periodically make larger re-adjustments in the expert layer necessary. As a result, in a model such as ours, validation loss may change in a wave-like pattern. This may have an impact on our choice of the ideal number of experts. For example, in models with one expert whose parameters map general text characteristics, DistilBERT model changes may require a reshuffling of components among experts less frequently, potentially leading to smoother curves. Further research would be needed to confirm this. This finding may also indicate that the overall model and performance may benefit from pre-training via Model Agnostic Meta Learning (Finn et al., 2017) [19] methods such as Reptile (Dou et al., 2019) [20] which would instantiate parameters in a way that achieves faster learning performance, thereby potentially smoothing fluctuations.

Overall the performance of our model suggests that it works well in our context. A strong limitation is that the outer-layer of our model architecture requires training data with an identifiable origin for any OOD data that we would like to predict.

6 Conclusion

This project demonstrates an inner-outer application of MoE which substantially increases the question-answering capabilities of a DistilBERT language model from baseline performance. We applied the general concomitant covariates principle of the MoE model both to individual heads within models (inner-layer) and as a design principle to structure multiple larger models (outer-layer), thereby increasing performance on a task with limited OOD data. Our experimentation has revealed an optimal arrangement of both our inner- and outer-layer experts. The decision to take over the task assignment of the gating feature in our outer-layer proved to be instrumental in an environment of OOD data scarcity where data origin is known. Overall, we believe that treating various elements of models as "experts" to be a potentially fruitful approach to future explorations into NLP applications, specifically for QA. Future designs may decrease fluctuations in prediction performance by pre-initializing weights for inner experts using Model Agnostic Meta Learning methods that increase learning speed or by reducing validation steps / adjusting the rules by which model checkpoints are saved to reduce overfitting.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017). Attention is all you need. arXiv preprint arXiv:1706.03762.

[2] Sanh, V., Debut, L., Chaumond, J., Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

[3] Jacobs, R. A., Jordan, M. I., Nowlan, S. J., Hinton, G. E. (1991). Adaptive mixtures of local experts. Neural computation, 3(1), 79-87.

[4] Gormley, I. C., Murphy, T. B. (2010). A mixture of experts latent position cluster model for social network data. Statistical methodology, 7(3), 385-405.

[5] Frühwirth-Schnatter, S., Pamminger, C., Weber, A., Winter-Ebmer, R. (2012). Labor market entry and earnings dynamics: Bayesian inference using mixtures-of-experts Markov chain clustering. Journal of Applied Econometrics, 27(7), 1116-1137.

[6] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q. (2015, May). Line: Large-scale information network embedding. In Proceedings of the 24th international conference on world wide web (pp. 1067-1077).

[7] Villani, A. C., Lemire, M., Fortin, G., Louis, E., Silverberg, M. S., Collette, C., ... Franchimont, D. (2009). Common variants in the NLRP3 region contribute to Crohn's disease susceptibility. Nature genetics, 41(1), 71.

[8] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538.

[9] Yao, B., Walther, D., Beck, D., Fei-Fei, L. (2009). Hierarchical mixture of classification experts uncovers interactions between brain regions. Advances in Neural Information Processing Systems, 22, 2178-2186.

[10] Rasmussen, C. E., Ghahramani, Z. (2002). Infinite mixtures of Gaussian process experts. Advances in neural information processing systems, 2, 881-888.

[11] Aljundi, R., Chakravarty, P., Tuytelaars, T. (2017). Expert gate: Lifelong learning with a network of experts. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3366-3375).

[12] Eigen, D., Ranzato, M. A., Sutskever, I. (2013). Learning factored representations in a deep mixture of experts. arXiv preprint arXiv:1312.4314.

[13] Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250.

[14] Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein D., Polosukhin, I., Kelcey M., Devlin J., Lee, K., Toutanova, K. N., Jones, L., Chang, M., Dai, A., Uszkoreit, J., Le, Q., Petrov, S. (2019). Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7, 453-466.

[15] Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., Suleman, K. (2016). Newsqa: A machine comprehension dataset. arXiv preprint arXiv:1611.09830.

[16] Lai, G., Xie, Q., Liu, H., Yang, Y., Hovy, E. (2017). Race: Large-scale reading comprehension dataset from examinations. arXiv preprint arXiv:1704.04683.

[17] Levy, O., Seo, M., Choi, E., Zettlemoyer, L. (2017). Zero-shot relation extraction via reading comprehension. arXiv preprint arXiv:1706.04115.

[18] Saha, A., Aralikatte, R., Khapra, M. M., Sankaranarayanan, K. (2018). DuoRC: Towards complex language understanding with paraphrased reading comprehension. arXiv preprint arXiv:1804.07927.

[19] Finn, C., Abbeel, P., Levine, S. (2017, July). Model-agnostic meta-learning for fast adaptation of deep networks. In International Conference on Machine Learning (pp. 1126-1135). PMLR.

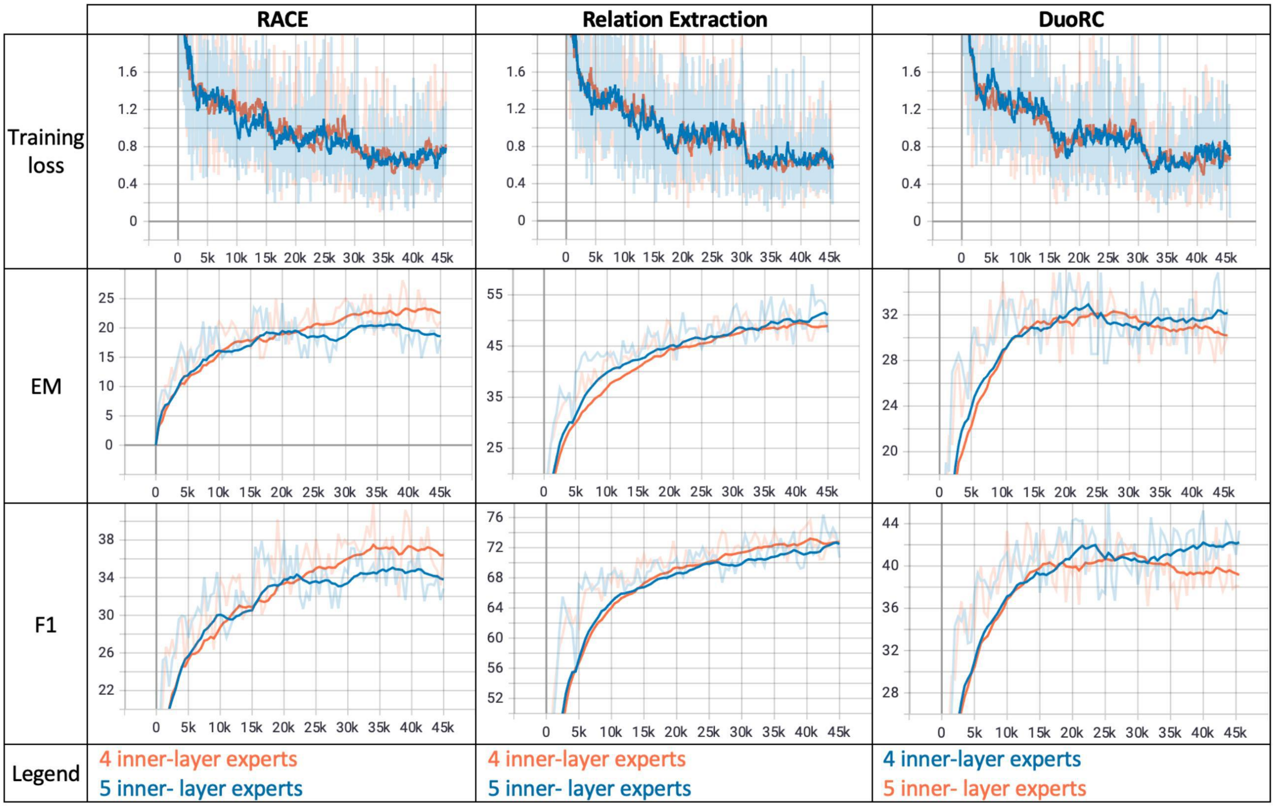
[20] Dou, Z. Y., Yu, K., Anastasopoulos, A. (2019). Investigating meta-learning algorithms for low-resource natural language understanding tasks. arXiv preprint arXiv:1908.10423.

A Appendix

Appendix 1: Number of answer-samples per dataset, split by training, validation, and test sets.

Dataset	**ID vs OOD**	**Abbreviation**	**Number of answer-samples**		
			Training	**Validation**	**Test**
SQuAD	ID	SQ	50,000	10,570	▨
Natural Questions	ID	NA	50,000	12,836	▨
NewsQA	ID	NE	50,000	4,212	▨
RACE	OOD	RA	127	128	419
Relation Extraction	OOD	RE	127	128	2,693
DuoRC	OOD	DU	127	128	1,248

Appendix 2: Training Loss, EM, and F1 graphs for 4 vs 5 expert versions of final models by dataset.


RACE
Relation Extraction
DuoRC
Training loss
EM
F1
Legend
0 5k 10k 15k 20k 25k 30k 35k 40k 45k
4 inner-layer experts
5 inner- layer experts
4 inner-layer experts
5 inner- layer experts
4 inner-layer experts
5 inner- layer experts